



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/799,740	03/12/2004	Barry Christopher Allyn	06576.105021 (MS# 150877.	9224
45979	7590	12/15/2006	EXAMINER	
PERKINS COIE LLP/MSFT P. O. BOX 1247 SEATTLE, WA 98111-1247			SINGH, RACHNA	
			ART UNIT	PAPER NUMBER
			2176	

DATE MAILED: 12/15/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

DETAILED ACTION

1. This action is responsive to communications: Amendments and Arguments filed on 09/26/06.
2. Claims 1-44, 46-53, and 55-67 are pending. Claims 1, 11, 19, 28, 36, 47, 55, 60, 66, and 67 are independent claims.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-2, 4-12, 14-20, 22-29, 31-38, 40-48, 50-62, and 64-67 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bryan et al, US 6,124,856, 09/26/00 (filed 04/24/96) in view of Chew, US 5,640,498, 06/17/97.

In reference to claims 1 and 11, Bryan teaches a method and apparatus for displaying modeless bar interfaces in a computer system which meets the preamble, "a

method in a computer system for displaying modeless windows". See title and abstract. Bryan discloses the following:

-Displaying an application window having a client area which meets the limitation,

"displaying an application window having a client area". See figure 2 and 3A-3D.

-Displaying a document window within the client area. See figure 3A where a user interface comprising a document viewing region 312 is shown which meets the

limitation, ***"within the client area, displaying a document window"***. See also column 5, lines 50-65.

-Representing modeless function interfaces or bars displayable by the application. See column 5, lines 38-67; column 6, lines 1-52; Figures 3A-3E. The modeless bar is anchored to the edge of the document window in figures 3A and 3E. The modeless bar contains titles of the menu options which are in a collapsed state. See figures 3A-3E and column 6, lines 1-51. Upon selection of a menu option, the modeless function object is initiated and the document currently open has its own copy of the modeless function interface. Compare to ***"displaying a modeless window in the document window anchored to an edge of the document window, the anchored modeless window having at least a collapsed and expanded states"***.

-Upon a selection of a menu option, the initiation of the modeless function object occurs.

The display tool enables simultaneous display of the different modeless bar interfaces within the same document view without obstructing other modeless windows which meets the limitation, ***“user input is received to the collapsed modeless window, determining a preferred position of the modeless window based upon its size in the expanded state, the preferred position calculated to prevent the modeless window from overlapping another modeless window”***. See figures 3A-3D and column 6, lines 46-52 and lines 1-45.

-The modeless windows are displayed in an expanded state without obstruction of other modeless windows and is anchored to the edge of a document window which meets the limitation, ***“expanding the collapsed modeless window so that it is in the expanded state and anchored to the edge of the document window based on the preferred position”***. See figure 3E and column 6.

-The modeless windows include information regarding the document such as a help function, table of content, review functions, etc which meets the limitation, ***“displaying information associated with the application within the expanded modeless window”***. See column 6, lines 17-35 and figure 3E.

Bryan does not expressly teach, ***when the modeless window is in the collapsed state, displaying its identifier without displaying its content***. Chew

discloses accessbars are anchored to the edge of a display where buttons 205, 207 refer to computer programs while in a minimized state which meets the limitation, ***when the modeless window is in the collapsed state, displaying its identifier without displaying its content.*** See figure 2b and columns 3-4.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include Chew's display of a title in a collapsed window state because it allows a user to easily identify the nature or type of information contained in the window even if it is minimized. See columns 3-4.

Bryan teaches receiving user input in order to expand a modeless window; however, he does not teach that when user input is received ***proximate or when user input is received that is not proximate to the expanded modeless window, collapsing the expanded modeless window so that it is in the collapsed state.***

The accessbars can be expanded to be displayed in a non-minimized state. Chew teaches an accessbar arbiter with an autohide function. An accessbar is anchored to the edge of a display and appears on the display at a given time. See column 1, lines 30-40. An accessbar can be a taskbar that is visible to a user interface element and informs a user of which tasks are active and have an active window. The taskbar is constructed so that it does not obscured by other open windows. The taskbar is anchored at fixed location on the user interface. See column 3, lines 10-35 and figure 2A. A taskbar or accessbar has an autohide property associated with it. An autohide property refers to when an accessbar is initially presented to a user in an invisible state as shown in figure 2E. When invisible, instead of displaying the accessbar, a "hotbar",

Art Unit: 2176

226, is displayed. The hotbar acts as a mechanism for displaying an accessbar in a visible manner. When a user moves a mouse cursor towards the hotbar, the accessbar reappears and becomes visible to the user. As the user moves away, the accessbar becomes invisible again. See columns 5, lines 25-67 and column 6, lines 1-4.

Compare to ***when user input is received that is proximate, expanding the window*** and ***when user input is received that is not proximate to the window, collapsing the window so that it is in its collapsed state.***

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. An accessbar or window that is consistently visible to a user may obstruct the views of other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claims 2 and 12, Bryan teaches updating information in the modeless window based on the document displayed. For example, one of the modeless windows is for a grammar check. If the document content has changed, then

Art Unit: 2176

the information in the modeless window would change to reflect changes made in the document that would require a new grammar check. See figure 3E.

In reference to claim 4, Bryan teaches that the modeless window can be displayed in the same document view. The document currently open by the word processing application may have its own copy of the modeless interface. See column 6, lines 9-16.

In reference to claims 5 and 14, Bryan teaches the modeless windows are related to the document window and provide functions such as grammar check functions, review functions, and merge functions for the document. A child window is simply a window related to a parent window.

In reference to claims 6 and 15, Bryan teaches displaying multiple modeless windows related to the document application. See figure 3E.

In reference to claim 7, Bryan teaches displaying multiple modeless windows in a manner so that there is no obstruction or overlap.

In reference to claims 8-9 and 16-17, Bryan teaches that upon a selection of a menu option, the initiation of the modeless function object expansion occurs. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Bryan teaches collapsing the

modeless window by depressing, with a mouse, the "done" button on the modeless window as depicted in figures 3B-3D. Expanding or collapsing a window entails changing the size of the window.

In reference to claims 10 and 18, Bryan does not teach expanding or collapsing a window based on the position of user input. However, Chew teaches an accessbar arbiter with an autohide function. An accessbar is anchored to the edge of a display and appears on the display at a given time. See column 1, lines 30-40. An accessbar can be a taskbar that is visible to a user interface element and informs a user of which tasks are active and have an active window. The taskbar is constructed so that it does not obscured by other open windows. The taskbar is anchored at fixed location on the user interface. See column 3, lines 10-35 and figure 2A. A taskbar or accessbar has an autohide property associated with it. An autohide property refers to when an accessbar is initially presented to a user in an invisible state as shown in figure 2E. When invisible, instead of displaying the accessbar, a "hotbar", 226, is displayed. The hotbar acts as a mechanism for displaying an accessbar in a visible manner. When a user moves a mouse cursor towards the hotbar, the accessbar reappears and becomes visible to the user. As the user moves away, the accessbar becomes invisible again. See columns 5, lines 25-67 and column 6, lines 1-4.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the

Art Unit: 2176

invention to prevent one screen object from negatively affecting another screen object.

An accessbar or window that is consistently visible to a user may obstruct the views of other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claims 19 and 28, Bryan teaches a method and apparatus for displaying modeless bar interfaces in a computer system. See title and abstract.

Compare to ***"a method in a computer system for displaying modeless windows"***.

Bryan discloses the following:

-Displaying an application window having a client area. See figure 2 and 3A-3D.

Compare to ***"displaying an application window having a client area"***.

-Displaying a document window within the client area. See figure 3A where a user interface comprising a document viewing region 312 is shown. See also column 5, lines 50-65. Compare to ***"within the client area, displaying a document window"***.

-Representing multiple modeless function interfaces or bars displayable by the application. See column 5, lines 38-67; column 6, lines 1-52; Figures 3A-3E. The

Art Unit: 2176

modeless bar is anchored to the edge of the document window in figures 3A and 3E.

The modeless window allows a user to interact with the window without requiring a user to open or close the window each time it is used. It can remain in a suspended state until resumed by the user. See column 1, lines 30-48. Modeless windows, unlike modal windows, do not require user action before the user can interact with the document window. Upon selection of a menu option, the modeless function object is initiated and the document currently open has its own copy of the modeless function interface. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view without obstructing other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. The modeless windows include information regarding the document such as a help function, table of content, review functions, etc. See column 6, lines 17-35 and figure 3E. Compare to ***“displaying a first and second modeless window in the document window that does not prevent functionality of the document window after being selected and that within it displays information associated with the application”.***

Bryan teaches that upon a selection of a menu option, the initiation of the modeless function object occurs. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view *without obstructing* other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Once a new object is added to the display, the various objects are rearranged so that they do not overlap. See column 7, lines 54-60. A user can manually resize the display

Art Unit: 2176

which would require resizing of all objects. Specifically, Bryan teaches if a new object is to be displayed, an area manager obtains the priority value for the displayed object and compares it to other display objects. If the priority of the new object is greater than the priority of another object, then the new object is inserted before the other object requiring the objects to be rearranged and moved. See column 7, lines 25-67 and figures 5A-5B.

Bryan teaches moving the first window or object if the priority of the new object is greater; however, Bryan does not expressly state moving the window in response to a window movement command. Chew teaches a system that governs the location of an accessbar by receiving requests for proposed locations. See abstract and column 5, lines 42-67

It would have been obvious to a person of ordinary skill in the art at the time of the invention the teachings of Chew's movement command in Bryan's system of displaying windows in order of priority because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. Thus it was desirable at the time of the invention to provide a means to move the location of windows in order to prevent conflict with other screen objects. See column 7, lines 25-67 of Bryan and column 5, lines 42-67 of Chew.

In reference to claims 20 and 29, Bryan teaches a user can double click on a menu item to initiate the display of a modeless function object.

In reference to claims 22 and 31, Bryan teaches that the modeless window can be displayed in the same document view. The document currently open by the word processing application may have its own copy of the modeless interface. See column 6, lines 9-16.

In reference to claim 23, Bryan teaches the modeless window is anchored to the edge of a document window. See figure 3E.

In reference to claims 24 and 32, Bryan teaches the modeless windows are related to the document window and provide functions such as grammar check functions, review functions, and merge functions for the document. A child window is simply a window related to a parent window.

In reference to claims 25-26 and 33-34, Bryan teaches that upon a selection of a menu option, the initiation of the modeless function object expansion occurs. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Bryan teaches collapsing the modeless window by depressing, with a mouse, the "done" button on the modeless window as depicted in figures 3B-3D. Expanding or collapsing a window entails changing the size of the window.

In reference to claims 27 and 35, Bryan does not teach expanding or collapsing a window based on the position of user input. However, Chew teaches an accessbar

Art Unit: 2176

arbiter with an autohide function. An accessbar is anchored to the edge of a display and appears on the display at a given time. See column 1, lines 30-40. An accessbar can be a taskbar that is visible to a user interface element and informs a user of which tasks are active and have an active window. The taskbar is constructed so that it does not obscured by other open windows. The taskbar is anchored at fixed location on the user interface. See column 3, lines 10-35 and figure 2A. A taskbar or accessbar has an autohide property associated with it. An autohide property refers to when an accessbar is initially presented to a user in an invisible state as shown in figure 2E. When invisible, instead of displaying the accessbar, a "hotbar", 226, is displayed. The hotbar acts as a mechanism for displaying an accessbar in a visible manner. When a user moves a mouse cursor towards the hotbar, the accessbar reappears and becomes visible to the user. As the user moves away, the accessbar becomes invisible again. See columns 5, lines 25-67 and column 6, lines 1-4.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. An accessbar or window that is consistently visible to a user may obstruct the views of other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or

windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claims 36 and 47, Bryan teaches a method and apparatus for displaying modeless bar interfaces in a computer system. See title and abstract.

Compare to ***"a method in a computer system for displaying modeless windows"***.

Bryan discloses the following:

-Displaying an application window having a client area. See figure 2 and 3A-3D.

Compare to ***"displaying an application window having a client area"***.

-Displaying a document window within the client area. See figure 3A where a user interface comprising a document viewing region 312 is shown. See also column 5, lines 50-65. Compare to ***"within the client area, displaying a document window"***.

-Representing modeless function interfaces or bars displayable by the application. See column 5, lines 38-67; column 6, lines 1-52; Figures 3A-3E. The modeless bar is anchored to the edge of the document window in figures 3A and 3E. The modeless bar contains titles of the menu options which are in a collapsed state. See figures 3A-3E and column 6, lines 1-51. Upon selection of a menu option, the modeless function object is initiated and expanded. See figure 3A-3E. The modeless windows include information regarding the document such as a help function, table of content, review

Art Unit: 2176

functions, etc. See column 6, lines 17-35 and figure 3E. Compare to ***“displaying in the document window a modeless window in an expanded state that displays information regarding the application”***.

Bryan does not expressly teach, ***collapsing the modeless window such that a title bar associated with the modeless window is displayed without displaying the information regarding the application***. Chew discloses accessbars are anchored to the edge of a display where buttons 205, 207 refer to computer programs while in a minimized state which meets the limitation, ***when the modeless window is in the collapsed state, displaying its identifier without displaying its content***. See figure 2b and columns 3-4.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include Chew's display of a title in a collapsed window state because it allows a user to easily identify the nature or type of information contained in the window even if it is minimized. See columns 3-4.

Bryan teaches receiving user input in order to expand a modeless window; however, he does not teach that collapsing the window ***when user input selects a display position of the document window that is not near the modeless window and the collapsed modeless window is expanded when user input selects a display position of the document window that is near the collapsed modeless window***.

Chew teaches an accessbar arbiter with an autohide function. An accessbar is anchored to the edge of a display and appears on the display at a given time. See column 1, lines 30-40. An accessbar can be a taskbar that is visible to a user interface element and informs a user of which tasks are active and have an active window. The taskbar is constructed so that it does not obscured by other open windows. The taskbar is anchored at fixed location on the user interface. See column 3, lines 10-35 and figure 2A. A taskbar or accessbar has an autohide property associated with it. An autohide property refers to when an accessbar is initially presented to a user in an invisible state as shown in figure 2E. When invisible, instead of displaying the accessbar, a "hotbar", 226, is displayed. The hotbar acts as a mechanism for displaying an accessbar in a visible manner. When a user moves a mouse cursor towards the hotbar, the accessbar reappears and becomes visible to the user. As the user moves away, the accessbar becomes invisible again. See columns 5, lines 25-67 and column 6, lines 1-4.

Compare to ***when user input selects a display position of the document window that is not near the modeless window and the collapsed modeless window is expanded when user input selects a display position of the document window that is near the collapsed modeless window.***

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object.

An accessbar or window that is consistently visible to a user may obstruct the views of other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claim 37, Chew teaches receiving user input via a mouse. See columns 5, lines 25-67 and column 6, lines 1-4. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. An accessbar or window that is consistently visible to a user may obstruct the views of other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claims 38 and 48, Bryan teaches updating information in the modeless window based on the document displayed. For example, one of the modeless windows is for a grammar check. If the document content has changed, then

the information in the modeless window would change to reflect changes made in the document that would require a new grammar check. See figure 3E.

In reference to claim 40, Bryan teaches that the modeless window can be displayed in the same document view. The document currently open by the word processing application may have its own copy of the modeless interface. See column 6, lines 9-16.

In reference to claims 41 and 50, Bryan teaches the modeless window is anchored to the edge of a document window. See figure 3E.

In reference to claims 42 and 51, Bryan teaches displaying multiple modeless windows related to the document application. See figure 3E.

In reference to claim 43, Bryan teaches displaying multiple modeless windows in a manner so that there is no obstruction or overlap.

In reference to claims 44 and 52, Bryan teaches that upon a selection of a menu option, the initiation of the modeless function object expansion occurs. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Bryan teaches collapsing the modeless window by depressing, with a mouse, the "done" button on the modeless window as

Art Unit: 2176

depicted in figures 3B-3D. Expanding or collapsing a window entails changing the size of the window.

In reference to claim 46, Bryan teaches the modeless windows are related to the document window and provide functions such as grammar check functions, review functions, and merge functions for the document. A child window is simply a window related to a parent window.

In reference to claim 53, Chew teaches receiving user input via a mouse. See columns 5, lines 25-67 and column 6, lines 1-4. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. An accessbar or window that is consistently visible to a user may obstruct the views of other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claims 55 and 60, Bryan teaches a method and apparatus for displaying modeless bar interfaces in a computer system. See title and abstract. Bryan discloses the following:

-Representing multiple modeless function interfaces or bars displayable by the application. See column 5, lines 38-67; column 6, lines 1-52; Figures 3A-3E. The modeless bar is anchored to the edge of the document window in figures 3A and 3E. The modeless window allows a user to interact with the window without requiring a user to open or close the window each time it is used. It can remain in a suspended state until resumed by the user. See column 1, lines 30-48. Modeless windows, unlike modal windows, do not require user action before the user can interact with the document window. Upon selection of a menu option, the modeless function object is initiated and the document currently open has its own copy of the modeless function interface. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view without obstructing other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. The modeless windows include information regarding the document such as a help function, table of content, review functions, etc. See column 6, lines 17-35 and figure 3E. Compare to ***"displaying a first and second modeless window that does not prevent functionality of the document window after being selected and that contains information about the computer program to the user, the modeless child window having a preferred location"***

-The display tool enables simultaneous display of the different modeless bar interfaces within the same document view without obstructing other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Compare to ***“anchoring the first modeless child window in a position that does not interfere with the preferred location of the second modeless child window”***.

Bryan teaches that upon a selection of a menu option, the initiation of the modeless function object occurs. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view *without obstructing* other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Once a new object is added to the display, the various objects are rearranged so that they do not overlap. See column 7, lines 54-60. A user can manually resize the display which would require resizing of all objects. Specifically, Bryan teaches if a new object is to be displayed, an area manager obtains the priority value for the displayed object and compares it to other display objects. If the priority of the new object is greater than the priority of another object, then the new object is inserted before the other object requiring the objects to be rearranged and moved which meets the limitation, ***in response to determining that the second modeless child window would overlap the first modeless child window, moving the first modeless child window to a new location in which the second modeless child window does not overlap the first child window***. See column 7, lines 25-67 and figures 5A-5B.

Bryan teaches moving the first window or object if the priority of the new object is greater; however, Bryan does not expressly state moving the window in response to a

Art Unit: 2176

window movement command. Chew teaches a system that governs the location of an accessbar by receiving requests for proposed locations. See abstract and column 5, lines 42-67

It would have been obvious to a person of ordinary skill in the art at the time of the invention the teachings of Chew's movement command in Bryan's system of displaying windows in order of priority because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. Thus it was desirable at the time of the invention to provide a means to move the location of windows in order to prevent conflict with other screen objects. See column 7, lines 25-67 of Bryan and column 5, lines 42-67 of Chew.

In reference to claims 56-57, Bryan does not teach closing and opening the modeless window responsive to user input; however, Chew teaches an accessbar arbiter with an autohide function. An accessbar is anchored to the edge of a display and appears on the display at a given time. See column 1, lines 30-40. An accessbar can be a taskbar that is visible to a user interface element and informs a user of which tasks are active and have an active window. The taskbar is constructed so that it does not obscured by other open windows. The taskbar is anchored at fixed location on the user interface. See column 3, lines 10-35 and figure 2A. A taskbar or accessbar has an autohide property associated with it. An autohide property refers to when an accessbar is initially presented to a user in an invisible state as shown in figure 2E. When invisible, instead of displaying the accessbar, a "hotbar", 226, is displayed. The hotbar acts as a mechanism for displaying an accessbar in a visible manner. When a

Art Unit: 2176

user moves a mouse cursor towards the hotbar, the accessbar reappears and becomes visible to the user. As the user moves away, the accessbar becomes invisible again.

See columns 5, lines 25-67 and column 6, lines 1-4.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. An accessbar or window that is consistently visible to a user may obstruct the views of other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claim 58, Bryan teaches both modeless windows are anchored. See figures 3A-3B.

In reference to claim 59, Chew teaches receiving user input via a mouse. See columns 5, lines 25-67 and column 6, lines 1-4.

In reference to claim 61, Bryan teaches updating information in the modeless window based on the document displayed. For example, one of the modeless windows is for a grammar check. If the document content has changed, then the information in

the modeless window would change to reflect changes made in the document that would require a new grammar check. See figure 3E.

In reference to claim 62, Bryan does not teach closing and opening the modeless window responsive to user input; however, Chew teaches an accessbar arbiter with an autohide function. An accessbar is anchored to the edge of a display and appears on the display at a given time. See column 1, lines 30-40. An accessbar can be a taskbar that is visible to a user interface element and informs a user of which tasks are active and have an active window. The taskbar is constructed so that it does not obscured by other open windows. The taskbar is anchored at fixed location on the user interface. See column 3, lines 10-35 and figure 2A. A taskbar or accessbar has an autohide property associated with it. An autohide property refers to when an accessbar is initially presented to a user in an invisible state as shown in figure 2E. When invisible, instead of displaying the accessbar, a "hotbar", 226, is displayed. The hotbar acts as a mechanism for displaying an accessbar in a visible manner. When a user moves a mouse cursor towards the hotbar, the accessbar reappears and becomes visible to the user. As the user moves away, the accessbar becomes invisible again. See columns 5, lines 25-67 and column 6, lines 1-4.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of Chew's autohide function in the system of Bryan's display of modeless windows because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. An accessbar or window that is consistently visible to a user may obstruct the views of

Art Unit: 2176

other accessbars or windows because there is no limit to the number of accessbars or windows that can appear on a display at any given time. Thus it was desirable at the time of the invention to provide an "autohide" feature with regards to accessbars or windows that did not need to be displayed at that time in order to prevent conflict with other screen objects. See abstract of Chew.

In reference to claim 63, Bryan teaches collapsing the window by pressing "done" as depicted in figures 3B-3D. Clicking on "done" will collapse the window such that it is not at the edge of the display window. See figures 3A and 3E.

In reference to claims 64-65, Chew teaches receiving user input via a mouse. See columns 5, lines 25-67 and column 6, lines 1-4.

In reference to claim 66, Bryan teaches a method and apparatus for displaying modeless bar interfaces in a computer system. See title and abstract. Compare to "**a computer system for displaying modeless windows**". Bryan discloses the following:

-Displaying an application window having a client area. See figure 2 and 3A-3D.

Compare to "**a window display system that displays a window having a client area**".

Art Unit: 2176

-Displaying a document window within the client area. See figure 3A where a user interface comprising a document viewing region 312 is shown. See also column 5, lines 50-65. Compare to ***“a second window display system that displays a document window within the client area”***.

-Representing modeless function interfaces or bars displayable by the application. See column 5, lines 38-67; column 6, lines 1-52; Figures 3A-3E. The modeless bar is anchored to the edge of the document window in figures 3A and 3E. The modeless window allows a user to interact with the window without requiring a user to open or close the window each time it is used. It can remain in a suspended state until resumed by the user. See column 1, lines 30-48. Modeless windows, unlike modal windows, do not require user action before the user can interact with the document window.

Compare to ***“a third window display system that displays a first modeless window that does not prevent functionality of the document window after being selected, displays the first child window anchored to the edge of the document window”***.

-When a modeless window has not yet been selected, it is in a collapsed state where only the menu name is displayed. See figure 3A, bar 308 and figures 3B-3E. See also column 6, lines 1-50. Compare to ***“when the modeless window is in the collapsed state, displaying its identifier without displaying its content”***.

Art Unit: 2176

-Upon a selection of a menu option, the initiation of the modeless function object occurs.

The display tool enables simultaneous display of the different modeless bar interfaces within the same document view without obstructing other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Compare to ***“determines a preferred position for the first modeless child window based upon its size of its open state, even when the first modeless window is in a collapsed state”***.

-The modeless windows include information regarding the document such as a help function, table of content, review functions, etc. See column 6, lines 17-35 and figure 3E. Compare to ***“a content display system that displays information associated regarding the application within the modeless child window”***.

Bryan teaches that upon a selection of a menu option, the initiation of the modeless function object occurs. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view *without obstructing* other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Once a new object is added to the display, the various objects are rearranged so that they do not overlap. See column 7, lines 54-60. A user can manually resize the display which would require resizing of all objects. Specifically, Bryan teaches if a new object is to be displayed, an area manager obtains the priority value for the displayed object and compares it to other display objects. If the priority of the new object is greater than the priority of another object, then the new object is inserted before the other object

Art Unit: 2176

requiring the objects to be rearranged and moved which meets the limitation, ***moves the first modeless window when a user moves a second modeless child window to a position that would overlap the first modeless child window.*** See column 7, lines 25-67 and figures 5A-5B.

Bryan teaches moving the first window or object if the priority of the new object is greater; however, Bryan does not expressly state moving the window in response to a window movement command. Chew teaches a system that governs the location of an accessbar by receiving requests for proposed locations. See abstract and column 5, lines 42-67.

It would have been obvious to a person of ordinary skill in the art at the time of the invention the teachings of Chew's movement command in Bryan's system of displaying windows in order of priority because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. Thus it was desirable at the time of the invention to provide a means to move the location of windows in order to prevent conflict with other screen objects. See column 7, lines 25-67 of Bryan and column 5, lines 42-67 of Chew.

In reference to claim 67, Bryan teaches a a method for displaying modeless bar interfaces. Bryan teaches:

- The display system includes modeless windows include information regarding the document in a word processing application program such as a help function, table of

Art Unit: 2176

content, review functions, etc. See column 6, lines 17-35 and figure 3E. Compare to ***“a window display system that displays a modeless child window containing information about the computer program to the user”***.

- The modeless bar is anchored to the edge of the document window in figures 3A and 3E. Compare to ***“a window attacher for anchoring the modeless child window to an edge of the display window”***.

-Upon a selection of a menu option by a user, the initiation of the modeless function object occurs. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view without obstructing other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Compare to ***“an opening process that opens the modeless child window responsive to input received from the user”***.

-A user can click on “Done” to close the modeless window. See figures 3B-3D. Compare to ***“a closing process that closes the child window responsive to other input received from the user”***.

-Upon a selection of a menu option, the initiation of the modeless function object occurs. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view without obstructing other modeless windows. See

Art Unit: 2176

figures 3A-3D and column 6, lines 46-52 and lines 1-45. Compare to ***“a preferred position process that determines a preferred position of the modeless child window based upon a size of its open state when the modeless window is in a collapsed state”***.

Bryan teaches that upon a selection of a menu option, the initiation of the modeless function object occurs. The display tool enables simultaneous display of the different modeless bar interfaces within the same document view *without obstructing* other modeless windows. See figures 3A-3D and column 6, lines 46-52 and lines 1-45. Once a new object is added to the display, the various objects are rearranged so that they do not overlap. See column 7, lines 54-60. A user can manually resize the display which would require resizing of all objects. Specifically, Bryan teaches if a new object is to be displayed, an area manager obtains the priority value for the displayed object and compares it to other display objects. If the priority of the new object is greater than the priority of another object, then the new object is inserted before the other object requiring the objects to be rearranged and moved which meets the limitation, ***moves the first modeless window when a user moves a second modeless child window to a position that would overlap the first modeless child window***. See column 7, lines 25-67 and figures 5A-5B.

Bryan teaches moving the first window or object if the priority of the new object is greater; however, Bryan does not expressly state moving the window in response to a window movement command. Chew teaches a system that governs the location of an

Art Unit: 2176

accessbar by receiving requests for proposed locations. See abstract and column 5, lines 42-67.

It would have been obvious to a person of ordinary skill in the art at the time of the invention the teachings of Chew's movement command in Bryan's system of displaying windows in order of priority because it was desirable at the time of the invention to prevent one screen object from negatively affecting another screen object. Thus it was desirable at the time of the invention to provide a means to move the location of windows in order to prevent conflict with other screen objects. See column 7, lines 25-67 of Bryan and column 5, lines 42-67 of Chew.

5. Claims 3, 13, 21, 30, 39 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bryan et al, US 6,124,856, 09/26/00 (filed 04/24/96) in view of Chew, US 5,640,498, 06/17/97, as applied to claims 1, 11, 19, 28, 36, 47, 55, and 60 above, and further in view of Bronson, US 5,305,435, 04/19/94.

In reference to claims 3 and 13, Bryan does not teach that the document window is adjacent to at least two of the sides of the modeless window; however, Bronson does. Bronson teaches a window system in which a window can be expanded and collapsed within a document window. When in an "expanded" state, the window has at least two sides adjacent to the document window. See figures 4 and 5 that show the collapsed and expanded state of a window in a document. See also columns 1-3. It would have been obvious to a person of ordinary skill in the art at the time of the invention to

Art Unit: 2176

incorporate the features of Bronson in the modeless window display system of Bryan because it was desirable to maximize screen display of windows in an application program regardless of the modality. Furthermore, having an "active" and "inactive" display mode (or expanded and collapsed mode) would maximize the display area for the document window. See columns 1-3 of Bronson.

In reference to claims 21 and 30, Bryan does not teach that the document window is adjacent to at least two of the sides of the modeless window; however, Bronson does. Bronson teaches a window system in which a window can be expanded and collapsed within a document window. When in an "expanded" state, the window has at least two sides adjacent to the document window. See figures 4 and 5 that show the collapsed and expanded state of a window in a document. See also columns 1-3. It would have been obvious to a person of ordinary skill in the art at the time of the invention to incorporate the features of Bronson in the modeless window display system of Bryan because it was desirable to maximize screen display of windows in an application program regardless of the modality. Furthermore, having an "active" and "inactive" display mode (or expanded and collapsed mode) would maximize the display area for the document window. See columns 1-3 of Bronson.

In reference to claims 39 and 49, Bryan does not teach that the document window is adjacent to at least two of the sides of the modeless window; however, Bronson does. Bronson teaches a window system in which a window can be expanded

Art Unit: 2176

and collapsed within a document window. When in an "expanded" state, the window has at least two sides adjacent to the document window. See figures 4 and 5 that show the collapsed and expanded state of a window in a document. See also columns 1-3. It would have been obvious to a person of ordinary skill in the art at the time of the invention to incorporate the features of Bronson in the modeless window display system of Bryan because it was desirable to maximize screen display of windows in an application program regardless of the modality. Furthermore, having an "active" and "inactive" display mode (or expanded and collapsed mode) would maximize the display area for the document window. See columns 1-3 of Bronson.

Response to Arguments

6. Applicant's arguments and amendments filed on 09/26/06 have been fully considered:

Applicant argues two main arguments with respect to the previous office action. Applicant first argues that Bryan's modeless bar is not equivalent to the Applicant's modeless window because Bryan displays nothing when the modeless bar is not selected. In other words, when the anchored window is not select, Bryan does not teach displaying its title or menu name. Examiner agrees that Bryan does not expressly state that the title bar is displayed when the modeless window is in a collapsed state. However, upon further review, Chew discloses this feature. Specifically, Chew discloses accessbars are anchored to the edge of a display where buttons 205, 207

Art Unit: 2176

refer to computer programs while in a minimized state. See figure 2b and columns 3-4. The accessbars can be expanded to be displayed in a non-minimized state. Please refer to rejections above.

Second, Applicant argues that the current invention moves the present location of the first modeless window if a window movement command from a user is received that causes the second modeless window to be moved to a position that would overlap a preferred location of the first modeless window; whereas the prior art, Chew, teaches moving the "second modeless window", not the first. Examiner agrees with Applicant's assertions. However, upon further review, Bryan teaches displaying multiple modeless bars without obstructing each other. Specifically, Bryan teaches if a new object is to be displayed, an area manager obtains the priority value for the displayed object and compares it to other display objects. If the priority of the new object is greater than the priority of another object, then the new object is inserted before the other object requiring the objects to be rearranged. See column 7, lines 25-67 and figures 5A-5B. Please refer to rejections above.

In view of the comments above, the rejection is maintained.

Conclusion

Art Unit: 2176

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Rachna Singh whose telephone number is 571-272-4099. The examiner can normally be reached on M-F (8:30AM-6:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



Rachna Singh
12/06/06